Paul D'Avilar Dennis Taylor

Installing Fedora Core 4, OpenSSL, MySQL, Apache, PHP, Snort, BASE

Document Version: Semi-Final

Operating System:
Fedora Core 4

Target Software:

1. Snort <2.4.3>
2. Apache <2.0.54>
3. OpenSSL <Current Version>
4. PHP <5.0.4>
5. MySQL <4.1.16>
6. PEAR Modules (you may have install some dependencies) :
    a. Image_Color <1.0.2>
    b. Image_Canvas <0.2.4>
    c. Image_Graph <0.7.1>
7. ADOdb <4.68>
8. BASE <1.2.1> - Basic Analysis and Security Engine, the replacement for ACID

Introduction:

This document was created to provide support for someone attempting to install Snort Intrusion Detection System, where the goal is to have robust logging and visualization support integrated with Snort detection engine. Please note this how-to is a very compact set of instructions on getting Snort up and running, for more detail information please visit the individual products sites, links are provided in the reference section of this manual. Additionally, the platform you are installing Snort on should be hardened as best as possible since any IDS is a potential target to an attacker.

Installing Fedora Core 4:

It seems that since the system will become a Snort IDS, the best form of installation for Fedora is a custom install. You want to select the minimum set of packages as possible, this prevent unnecessary services and application. Once the install is complete you should turn off additional services that are not needed in your environment. The system should be as secure/ harden as possible before installing Snort. It seems that the best way to get Snort up and running fast, it to install many of the needed packages (MySQL, PHP, SSL, etc) during the install process and then use the "yum" utility to update these applications. Here are some things to pay attention to during the install process:

1. Firewall:
    a. Enable firewall
    b. Select SSH, HTTP, and HTTPS
    c. SELinux – warn
2. Web Server – Apache
    a. Crypto-Utils
    b. Dstcache_auth_mysql
    c. Mod_perl

d. Mod_ssl
e. PHP
f. PHP_mysql
g. Webalizer

3. Database – MySQL
a. MyODBC
b. Mod_auth_mysql
c. Mysql-devel
d. Mysql-server
e. Perl-DBD-MySQL
f. PHP-mysql

Once installation is complete, you should check for updates:

- rpm –import /usr/share/rhn/RPM-GPG-KEY-fedora
- yum –y update
- chkconfig yum on – to turn on nightly updates
  - You can also use the chkconfig utility to turn off services that are not needed
  - httpd and mysqld services should be turn on.
    - verify:
      - chkconfig –list | grep ":on"
    - If services are not on, turn on:
      - chkconfig httpd on
      - service httpd start
      - chkconfig mysqld on
      - service mysqld start

It is best to create a user account for Snort; this account can be created during the install process. In you are creating the account after installation, first create a group for the Snort user, and then create the user account. This manual assumes that the group and user account "snort" were created:

- groupadd snort
- useradd –g snort snort

Additionally the Bastille-Linux site (http://www.bastille-linux.org/) has a very good utility/ program that may provide additional assistance to harden the system.

Preparing the NIC:

It is best to run the Snort IDS with a static IP address. To setup a static address with host name (Note: The information entered below is an example of what the file should contain, you should use information that represents your environment):

/etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0
BOOTPROTO=static
BROADCAST=10.10.1.255
IPADDR=10.10.1.35
NETMASK=255.255.255.0
NETWORK=10.10.1.0
ONBOOT=yes
TYPE=Ethernet
```

/etc/sysconfig/networking

```
NETWORKING=yes
HOSTNAME=snort
GATEWAY=10.10.1.1
```

/etc/hosts

```
10.10.1.35      snort.mydomain_name         snort
```

It is very important that outbound transmissions (data leaving the Snort IDS) be controlled or limited. This can be done in several ways:

1. Don't bind the an IP to the NIC
2. Run Snort using a passive listener
3. Use separate receive and transit interfaces
4. Firewall rules/ TCP wrappers, etc.

Configuring MySQL for snort:

MySQL install process creates initial accounts that do not have passwords. The initial root accounts passwords are empty, so anyone can connect to the MySQL server as root *without a password* and be granted all privileges. Two anonymous-user accounts are created, each with an empty username. The anonymous accounts have no passwords, so anyone can use them to connect to the MySQL server. You should insure that these accounts are assign passwords or deleted if not needed. For example, one way to assign passwords to the root accounts:

```
shell> mysql -u root
mysql> set password for 'root'@'localhost' = password('enter_passwd');
mysql> set password for 'root'@'your_machine_name' = password('enter_passwd');
```

Where 'enter_passwd' is the password for the root accounts and 'your_machine_name' is the name of the machine that MySQL is running on. For the anonymous account:

```
shell> mysql -u root -p
mysql> set password for ' '@'localhost' = password('enter_passwd');
mysql> set password for ' '@'your_machine_name' = password('enter_passwd');
```

Also one way to delete accounts; for example anonymous user accounts:

```
shell> mysql -u root -p
mysql> delete from mysql.user where User = '';
mysql> flush privileges;
```

Now that the initial accounts for secure, you can now prepare MySQL for Snort. You may want to disable remote connection from the root account to MySQL database and delete the 'test' database. Snort needs an account to access MySQL for outputting information:

```
shell> mysql -u root -p
mysql> create database snort;
mysql> grant insert, select on root.* to snort@localhost;
mysql> set password for snort@localhost=password('password_in_snort_conf');
mysql> grant create, insert, select, delete, update on snort.* to snort@localhost;
mysql> grant create, insert, select, delete, update on snort.* to snort;
```

The above creates the database 'snort' that will be use by Snort to log information. Please note that the password for the database is the same password use in the Snort configuration file, later in this manual.

Verifying Apache and PHP

The simplest way to test that PHP and Apache is working is to create a file (myfile.php) in the Apache Document root directory, on many systems the Apache Document root is located @ /var/www/html. The file should contain: "<?php phpinfo(); ?>". Next load up the browser: http://localhost/mytest.php, this should display information about packages and settings for PHP and Apache.

Configure Apache for SSL/ TLS:

BASE - Basic Security and Analysis Engine is a PHP GUI for analyzing Snort outputs. The utility requires a web server for serving pages/ displaying reports. This manual uses the Apache web server. During the installation process, Apache was installed with SSL/ TLS support. Once the Fedora installation process finishes and the system reboots, Apache running as httpd will start automatically every time the system boots. The web server is started with both http and https support. This manual assumes that BASE is the only process utilizing the web server. Keeping with the security mindset, default http (port 80) access to the web server is not allowed, only https access by configuring the Apache configuration file and firewall rules. Additionally, Apache will only serve BASE content. This configuration allows for BASE analysis to be view from any web browser and controls access to the server.

Things to keep in mind when running Apache

1. Limit/ stop users from setting up .htaccess files that can override security features explicitly configured
2. Protect server files by default
3. Limit Access to server and directories
4. Explicitly assign permissions to directories
5. Secure the communication channel
6. Monitor logs

The Apache configuration file, httpd.conf by default is located in /etc/httpd/conf directory. Within the httpd.conf file, other service specific configurations files such as ssl.conf, perl.conf are included/ imported from the /etc/httpd/conf.d directory. Therefore the httpd.conf file can be use to for general Apache settings and then use the ssl.conf file for SSL/TLS settings.

httpd.conf → The following settings should be included in the configuration file. Below represent basic settings for running the web server in relatively secure way. The web server can be further tweak to increase security and performance relative to a specific environment.

```
#-----------------------------------------------------------------------------------------------
# General Settings
#-----------------------------------------------------------------------------------------------
Listen host_ip:80 or 0.0.0.0:80 for all suitable interfaces
User apache
Group apache
UseCanonicalName off
ServerSignature off
HostnameLookups off
```

To prevent users from setting up .htaccess files that can override security features configured, use the 'AllowOverride' statement. Additionally, to prevent/ limit clients from walking through the entire filesystem, block access to the filesystem by default and use 'Directory' blocks to explicitly grant access to appropriate locations. See the BASE directory block later in this manual for an example of explicitly granting access.

```
#-----------------------------------------------------------------------------------------------
# General Access Control
#-----------------------------------------------------------------------------------------------
<Directory />
   Options None
   AllowOverride None
   Order Deny,Allow
   Deny from all
</Directory>
```

ssl.conf → Below are the basic SSL/ TLS settings for running Apache with medium to strong SSL/ TLS security. More or less restrictions can be added. Please refer to the Apache documentation site for more details on these settings and more.

```
#-----------------------------------------------------------------------------------------------
# SSL/ TLS Setting
#-----------------------------------------------------------------------------------------------
Listen host_ip:443
SSLEngine on
SSLMutex default
SSLProxyEngine off
SSLVerifyClient none
SSLCryptoDevice builtin
SSLOptions +StrictRequire
SSLSessionCacheTimeout  300
SSLPassPhraseDialog  builtin
SSLSessionCache         shmcb:/var/cache/mod_ssl/scache(512000)

AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl    .crl

SSLRandomSeed startup file:/dev/urandom  512
SSLRandomSeed connect file:/dev/urandom  512

SSLCertificateFile /etc/pki/tls/certs/localhost.crt
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key

SSLProtocol -all +TLSv1 +SSLv3
SSLCipherSuite HIGH:MEDIUM:!aNULL:+SHA1:+MD5:+HIGH:+MEDIUM

LogLevel warn
ErrorLog logs/ssl_error_log
TransferLog logs/ssl_access_log

<Files ~ "\.(cgi|shtml|phtml|php3?)$">
   SSLOptions +StdEnvVars
</Files>

<Directory "/var/www/cgi-bin">
   SSLOptions +StdEnvVars
</Directory>

SetEnvIf User-Agent ".*MSIE.*" \
       nokeepalive ssl-unclean-shutdown \
       downgrade-1.0 force-response-1.0
```

An additional layer of security is to use client certificates as well. Please refer to www.apache.org site for guides and how-tos for more information on configuring, securing, and running Apache web server.

Snort:

PCRE Install:

PCRE <current version>
Snort requires PCRE to function properly, therefore make sure that PCRE is installed. You can download PCRE from sourceforge.org. Once downloaded, extract PCRE and run:

- ./configure
- make install clean

Snort Install:

Should be root, or have permission to do the following. Download the latest stable version of Snort from sourceforge.org or snort.org. Once downloaded, extract Snort and run:

- ./configure --with-mysql {this is to enable Snort to log to MySQL database}
- make install

Snort expects to find the configuration file in either: /etc/snort or /usr/local/etc/snort, this manual uses: /etc/snort

- mkdir /etc/snort
- mkdir /etc/snort/rules
- mkdir /<available directory>/snort_backup
- mkdir /var/log/snort

Now copy the default rules and configuration files from the Snort installation directory to the recently created Snort directory: /etc/snort

- cd <snort installation directory>
- cp -R ./etc/snort/* /etc/snort/

Once the files are copied to the specified directory, the Snort configuration file must be modified for your specific environment. Specifically the following needed to be change:

- /etc/snort/snort.conf:

  var HOME_NET {should be set to your home network address}
  var RULE_PATH {should be: /etc/snort/rules}
  output database: log, mysql, user=snort password=whatever_you_want dbname=snort host=localhost {this instructs Snort to log to MySQL database}

Where the database Snort is logging to is "snort" with the user account "snort" and password "whatever_you_want" from the "localhost"

Verifying:

You should verify that the configuration file is correct and check that Snort is able to start by doing:

- /usr/local/bin/snort –c /etc/snort/snort.conf –g snort
  - –g – instruct snort to run as group 'snort' once initialization is complete.
  - You might need to specify the interface if snort cannot detect what interface on your system to use or if you want snort to use a specific interface. This can be done by specifying –i "iface"

Once you are confident that snort is running correctly, you can add a line in the rc.local file to make Snort start when the system starts. Included in the appendix section is a shell script for starting and stop Snort that was pulled off the web, use at your own risk.

- /usr/local/bin/snort –c /etc/snort/snort.config –g snort –i iface -u username

Schedule Snort rules to be downloaded automatically:

It is very important that you stay update with new Snort rules. Overtime, manual updating the rule-set can become very tedious. Oinkmaster is a Perl script that can be use to automatically update your Snort rule-set.

1. First register for the Sourcefire VRT Certified Rules on Snort.org.
2. Download and install the Oinkmaster utility. Please read the INSTALL file for information on installation and configuration of the Oinkmaster utility

Once Oinkmaster is configure correctly and completes successfully, the updating of rules can be schedule to run at specific times. One way to do this is to create a shell script with the specific syntax for running Oinkmaster. Then call the shell script using the cron daemon. The shell script allow for great flexibility.

SnortDailyUpdates.sh:

```
/usr/local/bin/oinkmaster.pl -C /usr/local/etc/oinkmaster.conf -C
/etc/snort/rules -b /etc/snort/backup 2>&1 | mail -s "oinkmaster"
snortadmin@mydomain.com
```

--The above statement is on the same line

This will run the Oinkmaster utility, creating a backup copy of the rules. Once the utility completes, an email will be sent with the results of the update.

```
Crontab:
```

```
30 7 * * * /<location of SnortDailyUpdates.sh>/SnortDailyUpdates.sh
```

This runs the SnortDailyUpdates.sh script 7:30AM everyday. Refer to the man pages for more information on crontabs.

Snort Schema:

Once MySQL is running and the 'snort' database is created, the snort schema/ table structures must be added to the database so Snort can accurately log to MySQL. The Snort install comes with pre-package scripts to create the snort schema. Do the following:

```
shell> mysql -u root –p < /<snort install/ make directory>/schemas/creat_mysql snort
```

Verify that the 'snort' database was create and check the database schema:

```
mysql> show databases;
mysql> use snort;
mysql> show tables;
```

At this point, Snort is installed, running, and logging to MySQL.

BASE – Basic Analysis and Security Engine:

Before installing BASE, some dependencies need to be installed first.

ADOdb:

BASE requires ADOdb to communicate with MySQL. ADOdb provides a performance-conscious database abstraction layer for PHP. ADOdb can be downloaded from sourceforge.org. Once downloaded, extract the package to a directory where base can access it. It is recommended but not required that ADOdb be placed in the Apache Document root.

PEAR – the PHP Extension and Application Repository Modules:

PEAR is usually installed during the PHP installation process. PEAR functions as CPAN for PHP. BASE requires several of the PEAR modules for rendering graphs. If not installed, install these modules by doing:

- pear install Image_Color
- pear install Image_Canvas
- pear install Image_Graph

Some dependencies may need to be installed before installing these modules.

BASE installation and Configuration:

BASE can be downloaded from sourceforge.org. Once downloaded, extract BASE to a directory that will be accessible from the web, for instance, /var/www/html/base, the Apache Document root, etc. Once extracted, Base's schema can be added to MySQL by doing:

```
shell> mysql -u root –p < /<base directory>/sql/create_base_tbls_mysql.sql snort
```

However it is best to use the BASE GUI to create the schema, this will be done very shortly.

Next modify the BASE configuration file to reflect your environment:

- cp base_conf.php.dist base_conf.php

Edit the base_conf.php:

```
$BASE_urlpath = "/base";
$DBlib_path = "/<location of ADOdb>"
$DBtype = "mysql"

$alert_dbname = "snort";
$alert_host = "localhost";
$alert_port = "";
$alert_user = "snort";
$alert_password = "use the same password from snort.conf";
```

The archive information database can also be configured:

```
$archive_dbname = "snort";
$archive_host = "localhost";
$archive_port = "";
$archive_user = "snort";
$archive_password = "use the same password from snort.conf";
```

Once the configuration file is saved, start BASE by opening a web browser and point to the directory. This manual assumes that BASE is installed in the Apache Document root:

http://localhost/base

Click on the setup link to create the MySQL schema if you have not done so already. Access to BASE need to be control by explicitly granting access in httpd.conf, you may need to do the following steps before accessing BASE from the above URL:

```
<Directory "/<BASE path>/base">
        Options Indexes FollowSymLinks
        AuthType Basic
        AuthName "SnortIDS"
        AuthUserFile /<path to base password file>/passwords
        Require user A_valid_base_user or Require valid-user
        AllowOverride AuthConfig
        Order Deny,Allow
        Allow from <wherever you want>
        SSLRequireSSL
</Directory>
```

Where ‘A_valid_base_user’ is a user in the BASE password file.

- mkdir <path to base passwords file>/passwords
- htpasswd –c /<path to base passwords file>/passwords/base {repeat this for more users}

References:

1. http://fedora.redhat.com
2. http://apache.org
3. http://www.snort.org
4. http://www.mysql.org
5. http://www.php.net
6. http://www.openssl.org
7. http://www.nist.gov
8. http://www.nsa.gov
9. * http://www.google.com

Appendix:

From the web USE AT OWN RISK

```
#!/bin/sh
#
# chkconfig: 2345 99 82
# description: Starts and stops the snort intrusion detection system
#
# config: /etc/snort/snort.conf
# processname: snort

# Source function library
. /etc/rc.d/init.d/functions

BASE=snort
DAEMON="-D"
INTERFACE="-i eth0"
CONF="/etc/snort/snort.conf"

# Check that $BASE exists.
[ -f /usr/local/bin/$BASE ] || exit 0

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

RETVAL=0
# See how we were called.
case "$1" in
  start)
        if [ -n "`/sbin/pidof $BASE`" ]; then
                echo -n $"$BASE: already running"
                echo ""
                exit $RETVAL
        fi
        echo -n "Starting snort service: "
        /usr/local/bin/$BASE $INTERFACE -c $CONF $DAEMON
        sleep 1
        action "" /sbin/pidof $BASE
        RETVAL=$?
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/snort
        ;;
  stop)
```

```
        echo -n "Shutting down snort service: "
        killproc $BASE
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/snort
        ;;
  restart|reload)
        $0 stop
        $0 start
        RETVAL=$?
        ;;
  status)
        status $BASE
        RETVAL=$?
        ;;
  *)
        echo "Usage: snort {start|stop|restart|reload|status}"
        exit 1
esac

exit $RETVAL
```